CLAIMS

What is claimed is:

1. A method of monitoring software executing on a trusted computing device comprising:
   generating in a protected partition on the trusted computing device baseline information pertaining to guest software in a guest virtual machine;
   storing the baseline information in a secure memory area;
   processing the guest software during runtime according to a predefined methodology to determine current runtime information; and
   comparing the current runtime information to the baseline information stored in the secure memory area to determine whether the guest software has been compromised.

2. The method according to Claim 1 wherein generating the baseline information further comprises performing a hash function on the guest software to obtain a hash value.

3. The method according to Claim 2 wherein performing the hash function on the guest software includes performing a hash function on one of each component of the guest software and a collection of components of the guest software.

4. The method according to Claim 2 wherein performing the hash function on the guest software to obtain the hash value further comprises at least one of performing the hash function on the guest software prior to execution to obtain an initial static baseline value and performing the hash function on the guest software immediately upon execution to obtain an initial runtime baseline value.

5. The method according to Claim 4 wherein processing the guest software during runtime according to a predefined methodology further comprises performing the hash function periodically on the guest software during runtime to obtain a current hash value.

6. The method according to Claim 5 wherein comparing the current runtime information to the baseline information further comprises comparing the current hash value to the baseline hash value.

7. The method according to Claim 1 wherein generating the baseline information comprises retrieving the baseline information from a storage location on the trusted computing device.

8. The method according to Claim 1 wherein storing the baseline information in the secure memory area further comprises storing the hash value in a trusted platform module ("TPM").

9. The method according to Claim 1 further comprising performing a secure launch of the trusted computing platform prior to generating the baseline information.

10. The method according to Claim 9 wherein storing the baseline information in the secure memory area further comprises storing the hash value in one of a TPM and a designated non-writable memory area.

11. The method according to Claim 9 further comprising executing at least a portion of the guest software in a designated non-writable memory area.

12. The method according to Claim 1 wherein the predefined methodology includes at least one of a checksum, MD5 and SHA1.

13. The method according to Claim 1 wherein the protected partition includes a root virtual machine.

14. A method of monitoring the integrity of a trusted computing device, comprising:

launching a protected partition and a guest virtual machine on the trusted computing device;

executing an integrity monitor in the protected partition and guest software in the guest virtual machine;

the integrity monitor processing the guest software in the guest virtual machine to generate a baseline hash value;

storing the baseline value in a secure memory area;

the integrity monitor periodically processing the guest software while executing to generate a current hash value; and

the integrity monitor comparing the baseline hash value in the secure memory area to the current hash value to determine whether the guest software has been compromised.

15. The method according to Claim 14 wherein storing the baseline value in a secure memory area includes storing the baseline value in at least one of a trusted platform module ("TPM") and a designated non-writable memory area.

16. The method according to Claim 14 further comprising processing and storing a value corresponding to the integrity monitor.

17. The method according to Claim 16 further comprising verifying the integrity monitor prior to comparing the baseline hash value to the current hash value.

18. The method according to Claim 14 wherein processing the guest software in the guest virtual machine to generate the baseline hash value includes retrieving the baseline hash value from a storage location.

19. The method according to Claim 14 wherein launching a protected partition includes launching a root virtual machine.

20. A system for monitoring software integrity, comprising:

a trusted computing device

a protected partition machine running on the trusted computing device;

a guest virtual machine running on the trusted computing device, the guest virtual machine including guest software;

a secure memory area on the trusted computing device; and

an integrity monitor executing within the protected partition, the integrity monitor capable of generating a baseline hash value for the guest software initially, and a current hash value for the guest software during runtime, the integrity monitor further capable of storing the baseline hash value in the secure memory area, the integrity monitor further capable of comparing the baseline hash value and the current hash value to determine if the guest software has been compromised.

21. The system according to Claim 20 wherein the secure memory area includes a trusted platform module ("TPM").

22. The system according to Claim 20 wherein the trusted computing device may calculate a hash value for the integrity monitor and store the hash value for the integrity monitor in the secure memory area.

23. The system according to Claim 22 wherein the hash value for the integrity monitor may be used to verify the integrity monitor prior to enabling the integrity monitor to access the baseline hash value stored in the secure memory area.

24. The system according to Claim 21 wherein the trusted computing device executes in Secure Execution Machine ("SMX") mode and the secure memory area includes one of the TPM and a designated non-writable memory area.

25. The system according to Claim 24 wherein a secure launch module may calculate a hash value for the integrity monitor and store the hash value for the integrity monitor in the secure memory area.

26. An article comprising a medium accessible by a trusted computing device, the medium having stored thereon instructions that, when executed by the trusted computing device, cause the trusted computing device to:
generate in a protected partition baseline information pertaining to components of guest software in a guest virtual machine;
store the baseline information in a secure memory area;
process the guest software during runtime according to a predefined methodology to determine current runtime information; and
compare the current runtime information to the baseline information stored in the secure memory area to determine whether the guest software has been compromised.

27. The article according to Claim 26 wherein the instructions, when executed by the trusted computing device, further cause the trusted computing device to perform a hash function on the guest software to obtain a hash value.

28. The article according to Claim 27 wherein perform a hash function on one of each component of the guest software and a collection of components of the guest software.

29. The article according to Claim 27 the instructions, when executed by the trusted computing device, further cause the trusted computing device to at least one of: perform the hash function on the guest software prior to execution to obtain an initial static baseline value and perform the hash function on the guest software immediately upon execution to obtain an initial runtime baseline value.

30. The article according to Claim 29 wherein the instructions, when executed by the trusted computing device, further cause the trusted computing device to perform the hash function periodically on the guest software during runtime to obtain a current hash value.

31. The article according to Claim 30 wherein the instructions, when executed by the trusted computing device, further cause the trusted computing device to compare the current hash value to the baseline hash value.

32. The article according to Claim 26 wherein the instructions, when executed by the trusted computing device, further cause the machine to retrieve the baseline information from a storage location on the trusted computing device.

33. The article according to Claim 26 wherein the instructions, when executed by the trusted computing device, further cause the trusted computing device to store the hash value in a trusted platform module ("TPM").

34. The article according to Claim 26 wherein the instructions, when executed by the trusted computing device, further cause the trusted computing device to perform a secure launch of the trusted computing platform prior to generating the baseline information.

35. The article according to Claim 34 wherein the instructions, when executed by the trusted computing device, further cause the trusted computing device to store the baseline value in one of a TPM and a designated non-writable memory area.

36. The article according to Claim 34 the instructions, when executed by the trusted computing device, further cause the trusted computing device to execute at least a portion of the guest software in a designated non-writable memory area.

37. The article according to Claim 26 wherein the protected partition includes a root virtual machine.